



# Lang !

## Recursos:

- operadores aritméticos: + e -
- operadores Lógicos: && (e lógico) e ! (negação)
- Comparações: < e == .
- Comandos: if-then-else, while e atribuições.
- Funções podem ser definidas e chamadas a qualquer momento !
- Um programa será uma coleção de funções e a função principal deverá ser chamada de main e não deve ter quaisquer argumentos !
- Não há comando explícito de retorno ! Toda função deve terminar com uma expressão e tal expressão é o resultado da função;
- Só pode-se operar entre valores de mesmo tipo !
- Int: (+,-,<,==); String (==, <), Bool (&&, ! , ==)

# Sintaxe Abstrata

```
Prgram ::= Func*
Func    ::= Type ID Params Cmd
Params  ::= Type ID (, Type ID)* | λ
Cmd ::= Cmd ; Cmd
      | IF Expr Cmd Cmd
      | WHILE Expr Cmd
      | VAR = Expr
      | Expr
Expr ::= Expr + Expr
      | Expr - Expr
      | Expr && Expr
      | Expr < Expr
      | Expr == Expr
      | ! Expr
      | ID | INT | true | false | STRING
Type ::= Int | Bool | String | Void
```

# Exemplo de um programa em Lang - Sintaxe Concreta

```
Int sum(Int x){  
    s = 1;  
    while(0 < x){  
        s = s + x;  
        x = x - 1;  
    }  
    s;  
}
```

# Exemplo de um programa em Lang - Sintaxe Concreta

```
Int sum(Int x){  
    s = 1;  
    while(0 < x){  
        s = s + x;  
        x = x - 1;  
    }  
    s;  
}
```

- O tipo da função está correto ?

# Exemplo de um programa em Lang - Sintaxe Concreta

```
Int sum(Int x){  
    s = 1;  
    while(0 < x){  
        s = s + x;  
        x = x - 1;  
    }  
    s;  
}
```

- O tipo da função está correto ?
- O Tipo da expressão  $x = x - 1$  está correto ?

# Exemplo de um programa em Lang - Sintaxe Concreta

```
Int sum(Int x){  
    s = 1;  
    while(0 < x){  
        s = s + x;  
        x = x - 1;  
    }  
    s;  
}
```

- O tipo da função está correto ?
- O Tipo da expressão  $x = x - 1$  está correto ?
- Existe algum tipo para a variável  $s$  ?



# So many (difficult) questions !

- O tipo da função está correto ? (*TypeChecking*)
- Para responder a essa pergunta usamos, assumimos que x tem tipo Int e verificamos que o retorno da função será de fato do tipo Int.

# So many (difficult) questions !

- O tipo da função está correto ? (*TypeChecking*)
- Para responder a essa pergunta usamos, assumimos que  $x$  tem tipo `Int` e verificamos que o retorno da função será de fato do tipo `Int`.
- O Tipo da expressão  $x = x - 1$  está correto ? (*TypeChecking*)
- Assumindo que  $x$  tem tipo `Int`, verificamos se o resultado do lado direito é compatível com a variável do lado esquerdo.

# So many (difficult) questions !

- O tipo da função está correto ? (*TypeChecking*)
- Para responder a essa pergunta usamos, assumimos que  $x$  tem tipo `Int` e verificamos que o retorno da função será de fato do tipo `Int`.
- O Tipo da expressão  $x = x - 1$  está correto ? (*TypeChecking*)
- Assumindo que  $x$  tem tipo `Int`, verificamos se o resultado do lado direito é compatível com a variável do lado esquerdo.
- Existe algum tipo para a variável  $s$  ? (*TypeInference*)
- O único tipo para  $s$  que faz com que o programa fique correto é `Int`.

## Considere agora esse programa !

```
Int mystery(Int x){  
    y = false;  
    if(y){  
        x = "nada";  
    }  
    else{  
        x = x + 1;  
    }  
    x;  
}
```

- O tipo da função está correto ?

# Considere agora esse programa !

```
Int mystery(Int x){  
    y = false;  
    if(y){  
        x = "nada";  
    }  
    else{  
        x = x + 1;  
    }  
    x;  
}
```

- O tipo da função está correto ?
- Quando podemos considerar um programa correto ? (Para isso precisamos de uma especificação.)

## Considere agora esse programa !

```
Int mystery(Int x){  
    y = false;  
    if(y){  
        x = "nada";  
    }  
    else{  
        x = x + 1;  
    }  
    x;  
}
```

- O tipo da função está correto ?
- Quando podemos considerar um programa correto ? (Para isso precisamos de uma especificação.)
- O que é correto ? (Tudo aquilo que atende a uma dada especificação pode ser dito correto em relação à especificação)

## Considere agora esse programa !

```
Int mystery(Int x){  
    y = false;  
    if(y){  
        x = "nada";  
    }  
    else{  
        x = x + 1;  
    }  
    x;  
}
```

- O tipo da função está correto ?
- Quando podemos considerar um programa correto ? (Para isso precisamos de uma especificação.)
- O que é correto ? (Tudo aquilo que atende a uma dada especificação pode ser dito correto em relação à especificação)
- É sempre possível descobrir se um programa está correto ? (Infelizmente não ! mas na dúvida diga que não ;-)

# E quanto a este outro !

```
Bool mystery2(){  
    z = z and !z;  
    z;  
}
```



# As regras do jogo !

- Para julgar se um programa, ou uma parte dele, está correto precisamos de um modelo de como um programa devia se comportar
- Em geral nos interessa que o programa não faça nada que possa colocar a máquina que o executa em um estado de erro !
- Além disso, só porque podemos , também proibimos certas operações as quais julgamos serem "maus habitos" na linguagem.
- Finalmente especificamos um conjunto de regras que dizem como um "bom programinha" deveria se comportar

# Sistemas de tipo- Formato das regras

$\frac{\text{Premissas}}{\text{conclusao}}$  Nome da regra

- Há um ambiente de tipos  $\Gamma$ , um mapeamento de nomes para tipos que registra tipo de cada identificador (nomes de variáveis, de funções, de classes e etc) no programa.
- Dizemos que de  $\Gamma$  podemos deduzir o um tipo de alguma coisa do seguinte modo:  $\Gamma \vdash t : \tau$ , onde  $\tau$  é um tipo qualquer e  $t$  é um fragmento (termo) do programa.

## Algumas regras para Lang !

$$\frac{(x, \tau) \in \Gamma}{\Gamma \vdash \text{var} : \tau} \text{ var}$$

$$\frac{}{\Gamma \vdash \text{INT} : \text{Int}} \text{ int}$$

$$\frac{}{\Gamma \vdash \text{BOOL} : \text{Bool}} \text{ bool}$$

$$\frac{}{\Gamma \vdash \text{STRING} : \text{String}} \text{ str}$$

# Algumas regras para Expressões !

$$\frac{\Gamma \vdash x_1 : Int \quad \Gamma \vdash x_2 : Int}{\Gamma \vdash x_1 + x_2 : Int} \text{ var}$$

$$\frac{\Gamma \vdash x_1 : Int \quad \Gamma \vdash x_2 : Int}{\Gamma \vdash x_1 - x_2 : Int} \text{ var}$$

$$\frac{\Gamma \vdash x_1 : Bool \quad \Gamma \vdash x_2 : Bool}{\Gamma \vdash x_1 \&\& x_2 : Bool} \text{ var}$$

$$\frac{\Gamma \vdash x_1 : Int \quad \Gamma \vdash x_2 : Int}{\Gamma \vdash x_1 == x_2 : Bool} \text{ var}$$

## Algumas regras comandos!

$$\frac{\Gamma \vdash E : \textit{Bool} \quad \Gamma \vdash C : \tau}{\Gamma \vdash (\textit{While } E \ C) : \tau} \textit{var}$$